COPY OF PAPERS
ORIGINALLY FILED

#7
DRWG
4-27-00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Fan, Jason C.; Jogalekar, Prasad P.; Bannai, Vinay K.
Assignee:	Luminous Networks, Inc.
Title:	Routing Switch Detecting Change In Session Identifier Before Reconfiguring Routing Table
Serial No.:	09/519,441
Filing Date:	March 3, 2000
Examiner:	Unknown
Group Art Unit:	2739
Docket No.:	M-8564 US

CIPE
FEB 27 2002
PATENT & TRADEMARK OFFICE

San Jose, Califor
February 15, 2(

RECEIVED
MAR 1 1 2002
Technology Center 2600

Attn: Official Draftsperson
COMMISSIONER FOR PATENTS
Washington, D. C. 20231

SUBMISSION OF FORMAL DRAWINGS

Dear Sir:

Applicants submit thirteen (13) sheets of formal drawings, consisting of Figures 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12A, 12B, in the above-named application. If there are any questions regarding these drawings, please call the undersigned at (408) 453-9200.

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Washington, D.C. 20231, on February 15, 2002.

Attorney for Applicant(s) — Date of Signature: 2/15/02

Respectfully submitted,

Brian D. Ogonowsky
Attorney for Applicant(s)
Reg. No. 31,988

LAW OFFICES OF
SKJERVEN MORRILL
MacPHERSON LLP

25 METRO DRIVE
SUITE 700
SAN JOSE, CA 95110
(408) 453-9200
FAX (408) 453-7979

APPROVED O.G. FIG.
BY
DRAFTSMAN
CLASS SUBCLASS

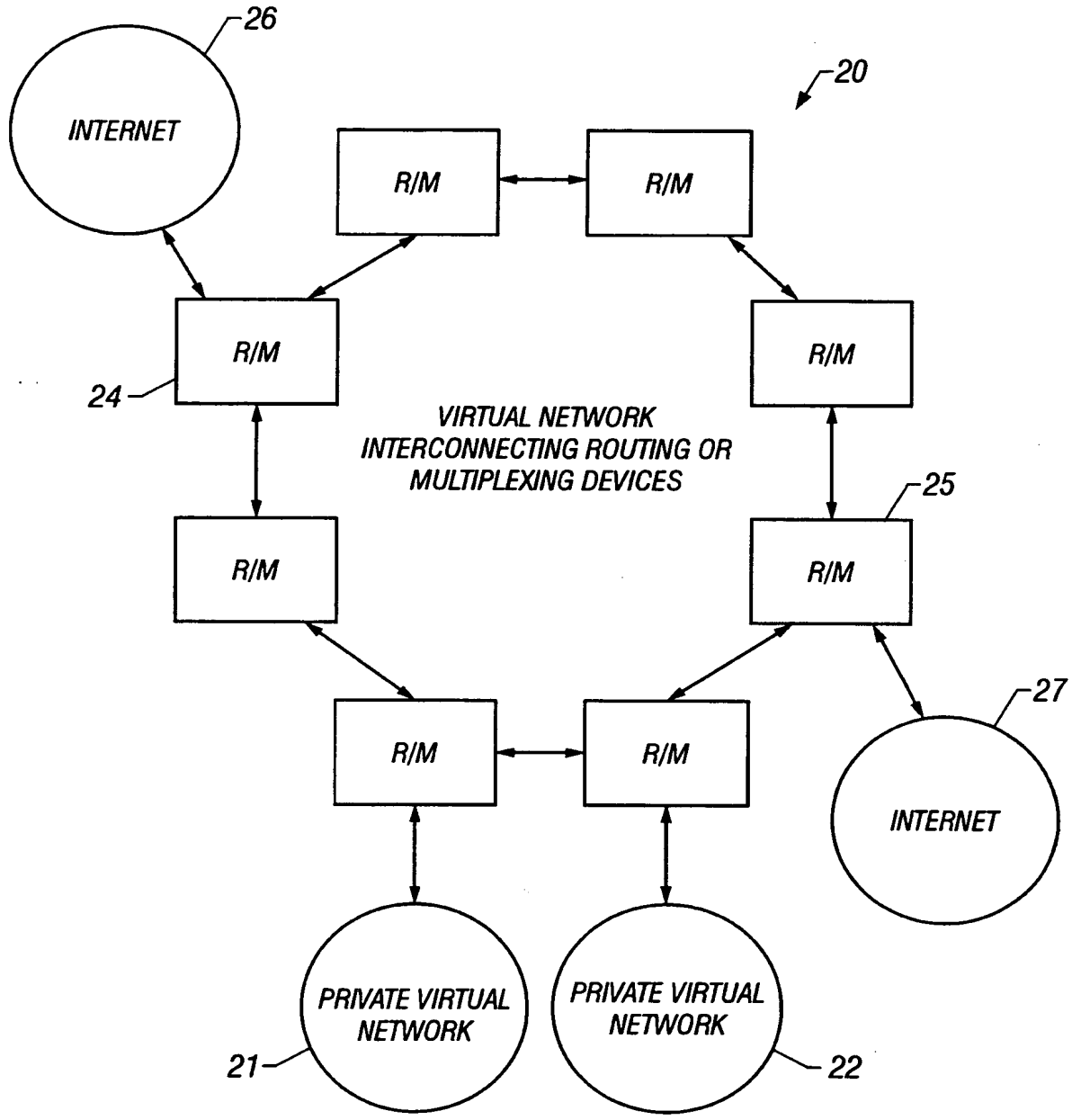

FIG. 1

APPROVED O.G. FIG.
BY

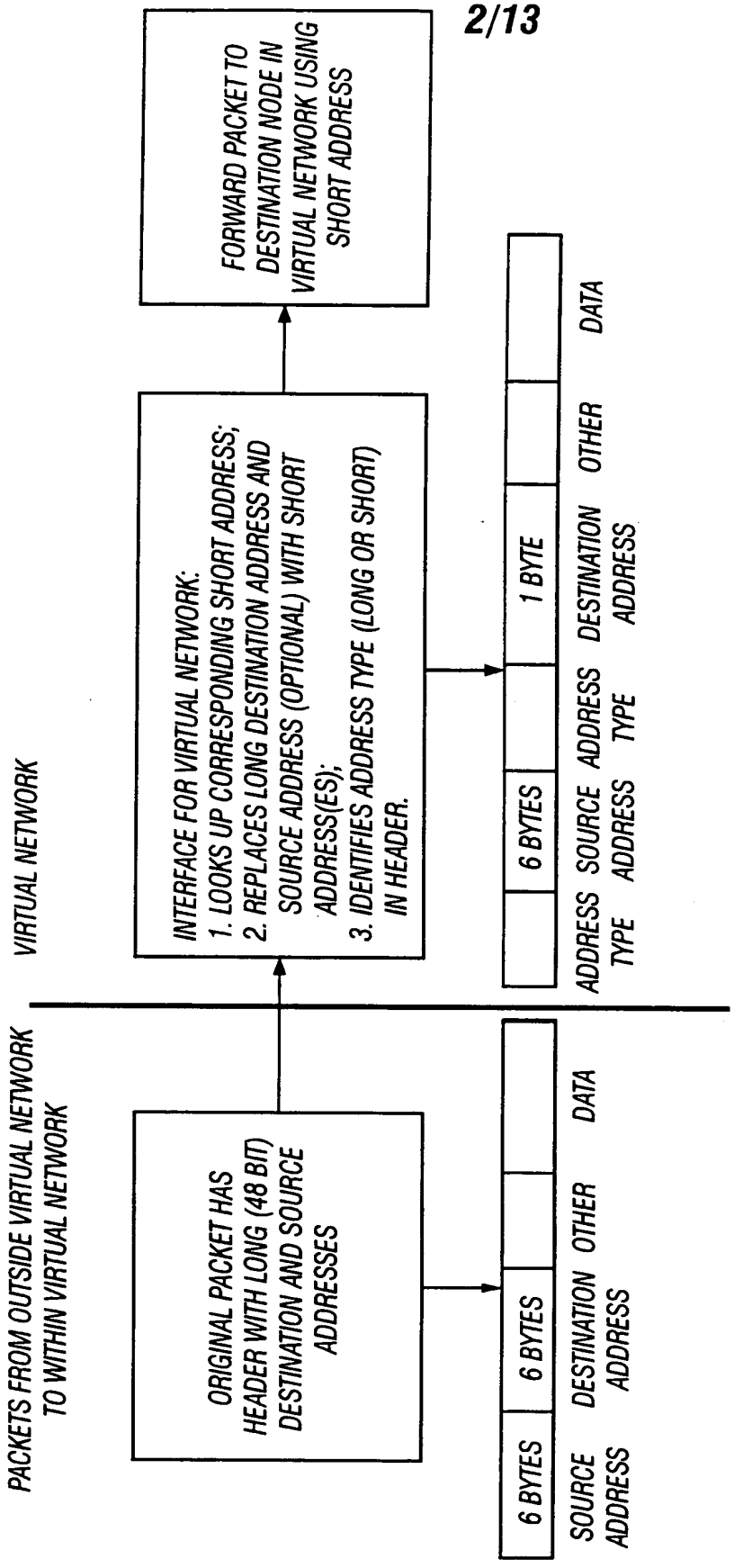

FIG. 2

APPROVED O.G. FIG.
BY
DATE

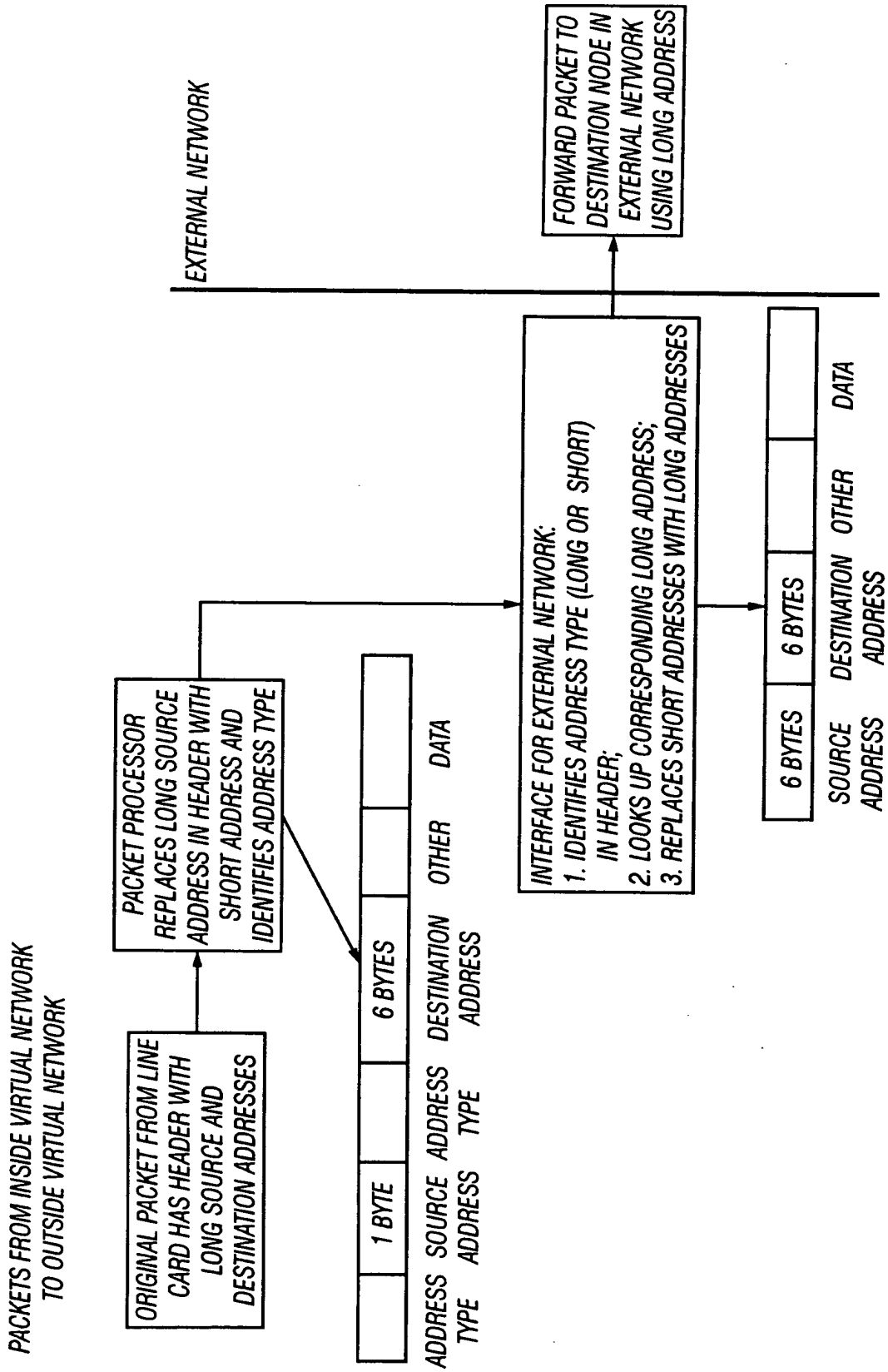

FIG. 3

APPROVED O.G. FIG.
BY
DIRECTOR

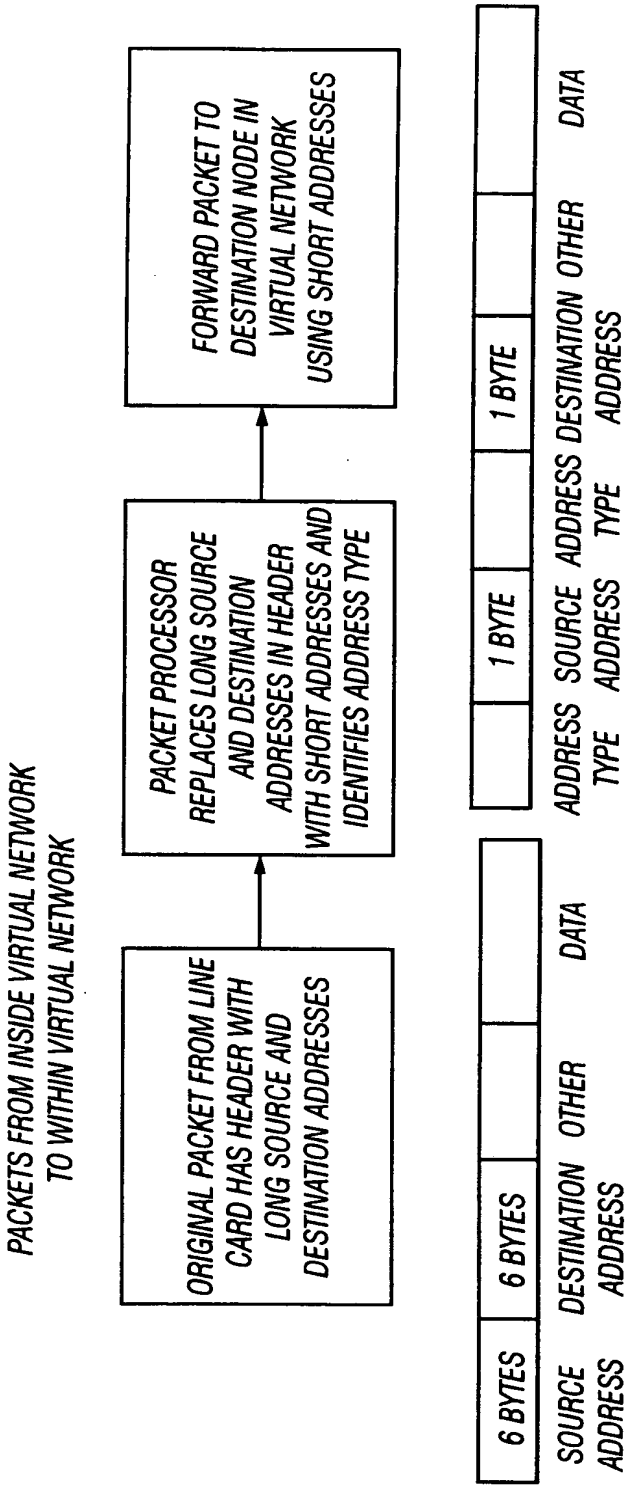

FIG. 4

APPROVED O.G. FIG.
BY

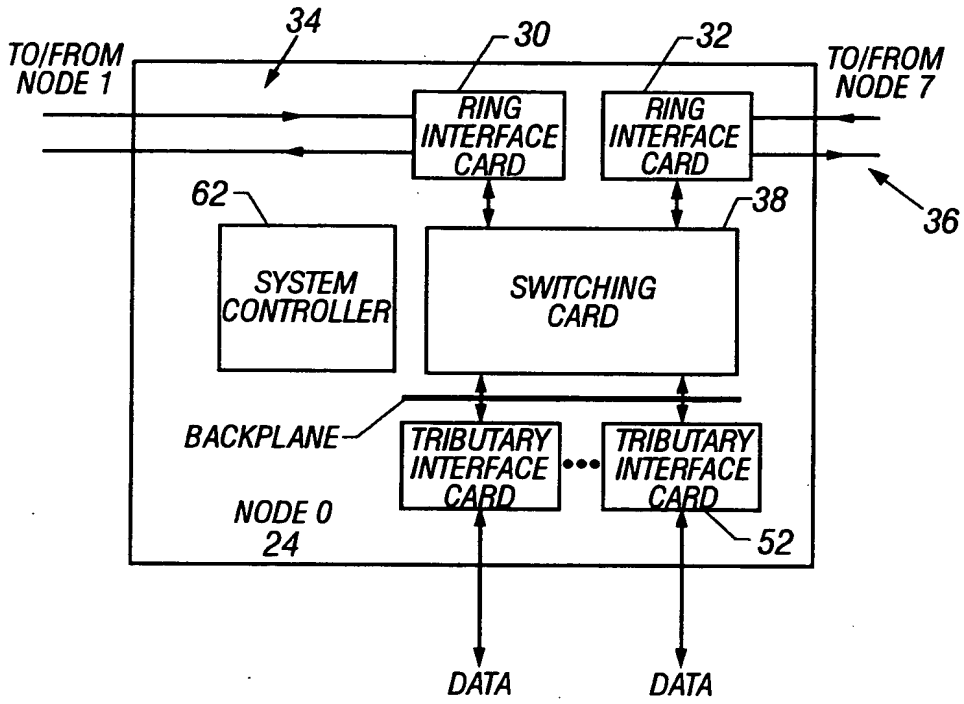

FIG. 5

APPROVED O.G. FIG.
BY
DRAFTSMAN

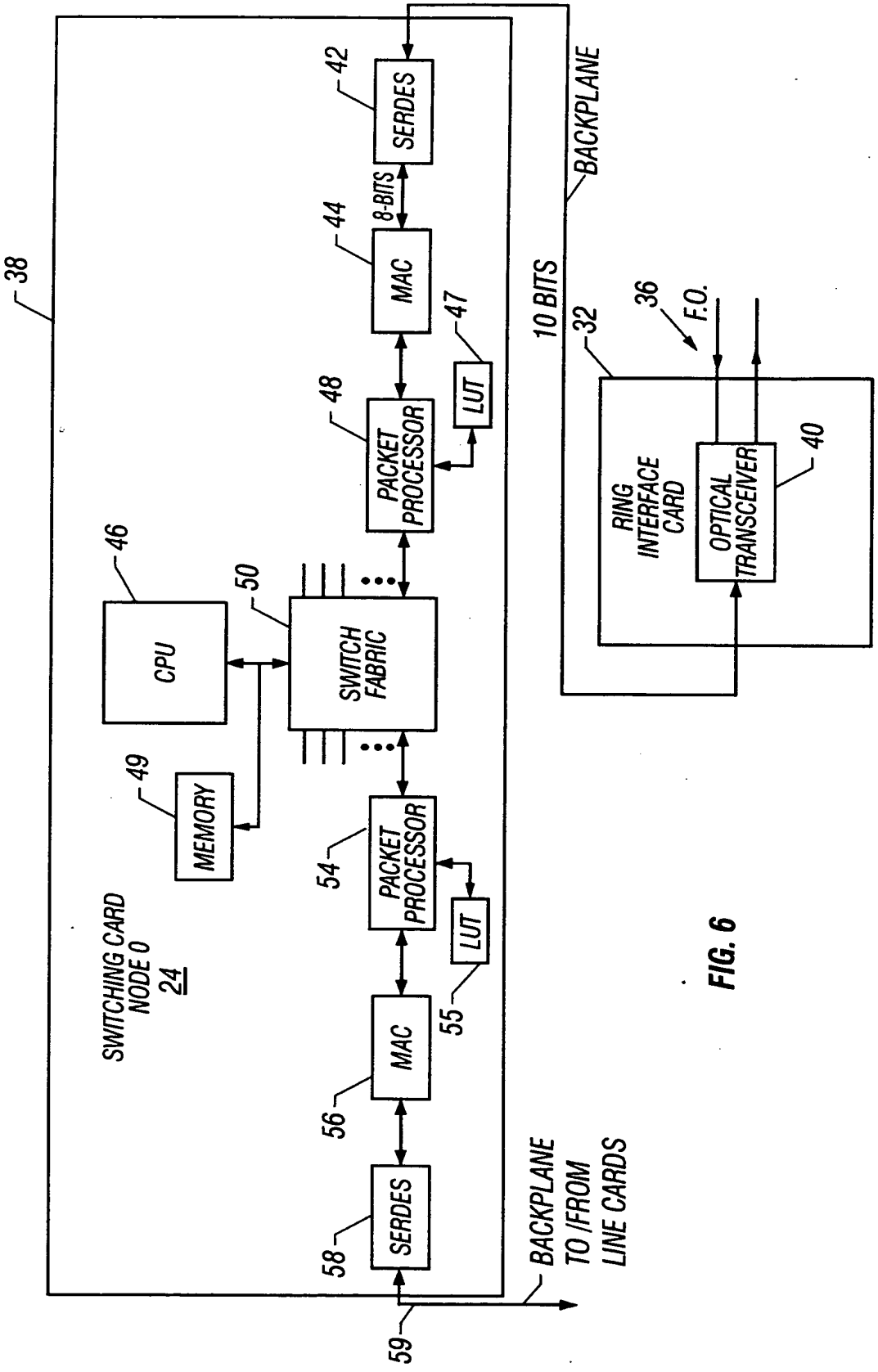

FIG. 6

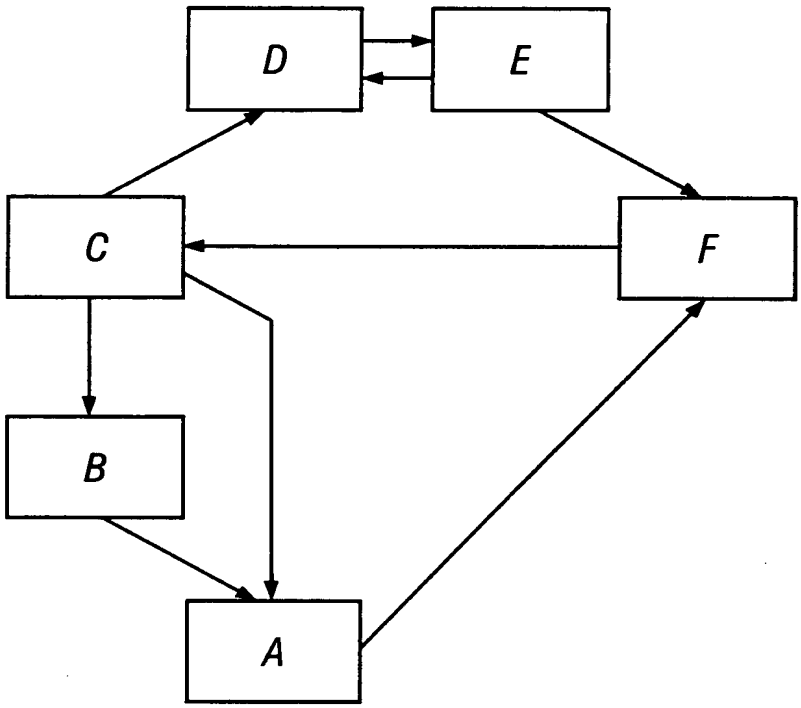

FIG. 7

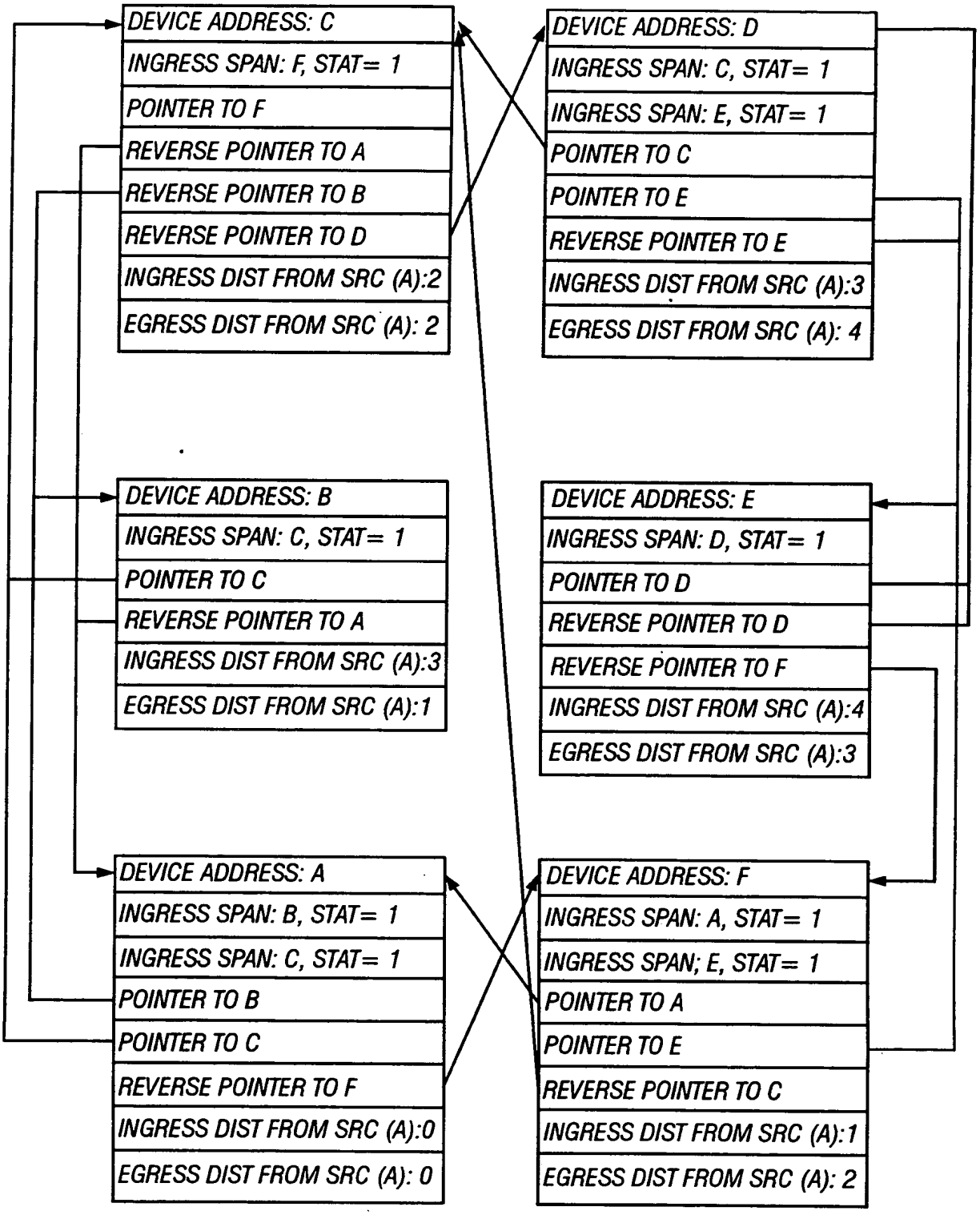
DEVICE ADDRESS: C
INGRESS SPAN: F, STAT= 1
POINTER TO F
REVERSE POINTER TO A
REVERSE POINTER TO B
REVERSE POINTER TO D
INGRESS DIST FROM SRC (A):2
EGRESS DIST FROM SRC (A): 2
DEVICE ADDRESS: D
INGRESS SPAN: C, STAT= 1
INGRESS SPAN: E, STAT= 1
POINTER TO C
POINTER TO E
REVERSE POINTER TO E
INGRESS DIST FROM SRC (A):3
EGRESS DIST FROM SRC (A): 4
DEVICE ADDRESS: B
INGRESS SPAN: C, STAT= 1
POINTER TO C
REVERSE POINTER TO A
INGRESS DIST FROM SRC (A):3
EGRESS DIST FROM SRC (A):1
DEVICE ADDRESS: E
INGRESS SPAN: D, STAT= 1
POINTER TO D
REVERSE POINTER TO D
REVERSE POINTER TO F
INGRESS DIST FROM SRC (A):4
EGRESS DIST FROM SRC (A):3
DEVICE ADDRESS: A
INGRESS SPAN: B, STAT= 1
INGRESS SPAN: C, STAT= 1
POINTER TO B
POINTER TO C
REVERSE POINTER TO F
INGRESS DIST FROM SRC (A):0
EGRESS DIST FROM SRC (A): 0
DEVICE ADDRESS: F
INGRESS SPAN: A, STAT= 1
INGRESS SPAN; E, STAT= 1
POINTER TO A
POINTER TO E
REVERSE POINTER TO C
INGRESS DIST FROM SRC (A):1
EGRESS DIST FROM SRC (A): 2

FIG. 8

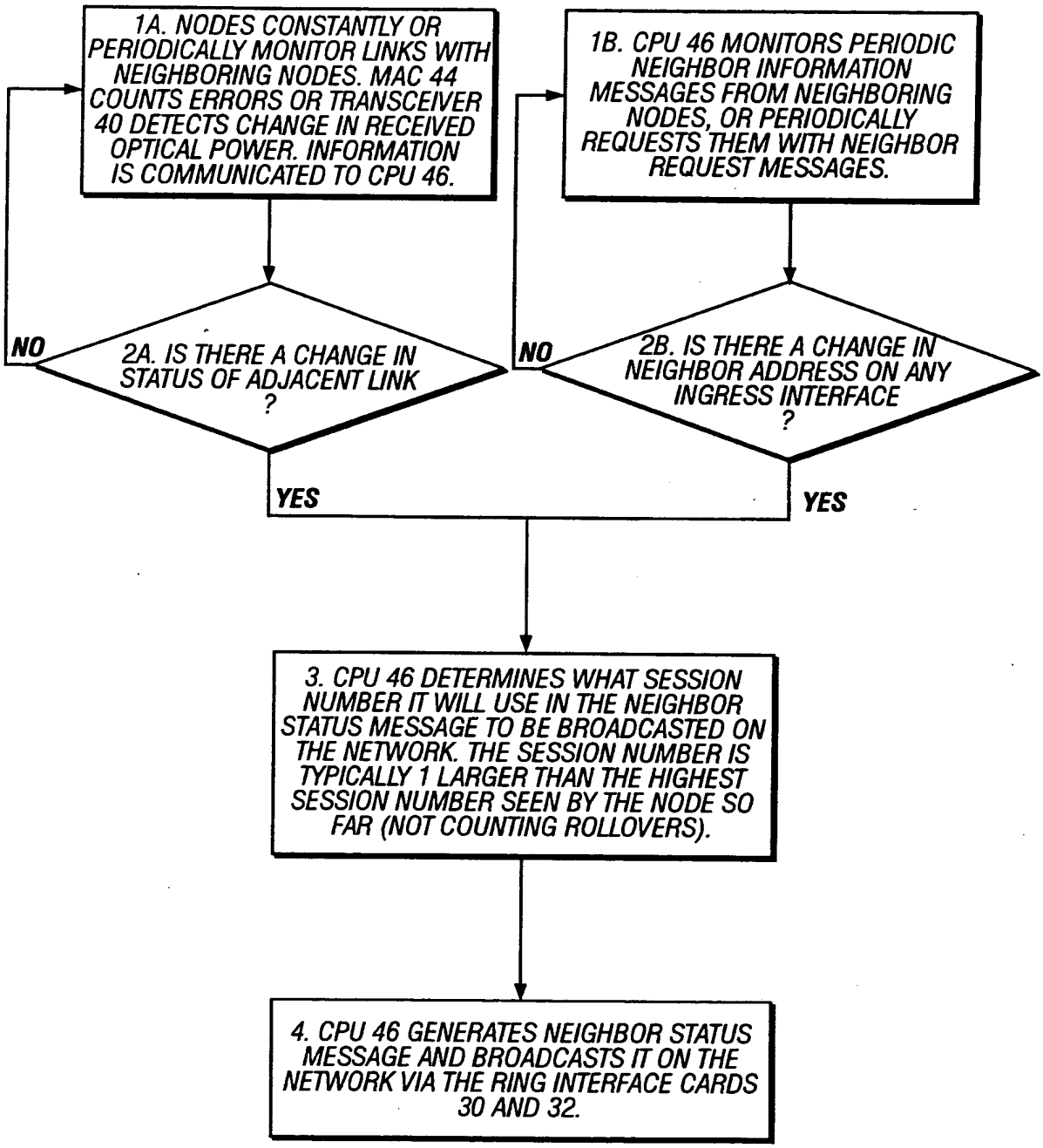
1A. NODES CONSTANTLY OR PERIODICALLY MONITOR LINKS WITH NEIGHBORING NODES. MAC 44 COUNTS ERRORS OR TRANSCEIVER 40 DETECTS CHANGE IN RECEIVED OPTICAL POWER. INFORMATION IS COMMUNICATED TO CPU 46.
1B. CPU 46 MONITORS PERIODIC NEIGHBOR INFORMATION MESSAGES FROM NEIGHBORING NODES, OR PERIODICALLY REQUESTS THEM WITH NEIGHBOR REQUEST MESSAGES.
2A. IS THERE A CHANGE IN STATUS OF ADJACENT LINK ?
NO
2B. IS THERE A CHANGE IN NEIGHBOR ADDRESS ON ANY INGRESS INTERFACE ?
NO
YES
YES
3. CPU 46 DETERMINES WHAT SESSION NUMBER IT WILL USE IN THE NEIGHBOR STATUS MESSAGE TO BE BROADCASTED ON THE NETWORK. THE SESSION NUMBER IS TYPICALLY 1 LARGER THAN THE HIGHEST SESSION NUMBER SEEN BY THE NODE SO FAR (NOT COUNTING ROLLOVERS).
4. CPU 46 GENERATES NEIGHBOR STATUS MESSAGE AND BROADCASTS IT ON THE NETWORK VIA THE RING INTERFACE CARDS 30 AND 32.

FIG. 9

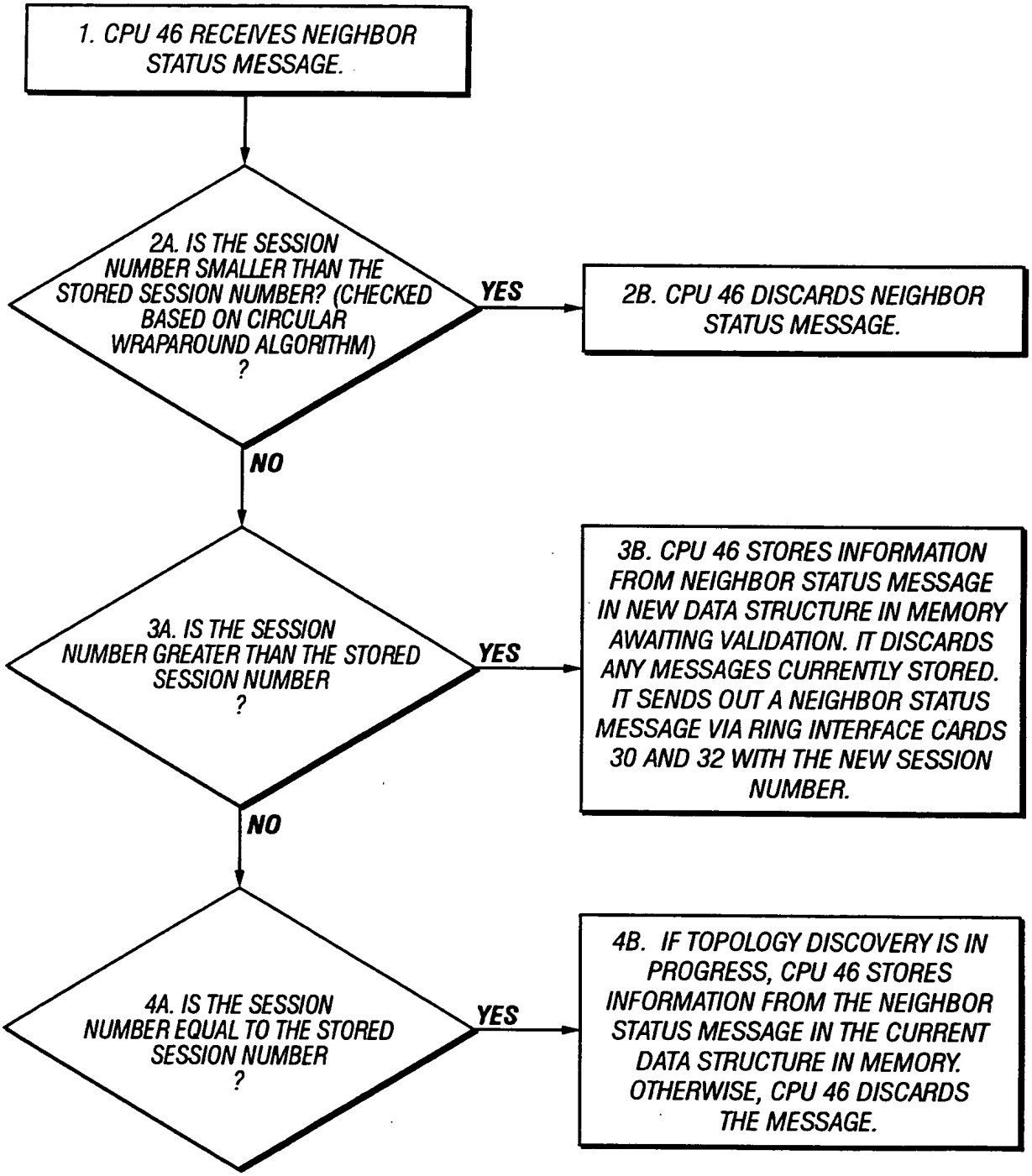
1. CPU 46 RECEIVES NEIGHBOR STATUS MESSAGE.
2A. IS THE SESSION NUMBER SMALLER THAN THE STORED SESSION NUMBER? (CHECKED BASED ON CIRCULAR WRAPAROUND ALGORITHM) ?
YES
2B. CPU 46 DISCARDS NEIGHBOR STATUS MESSAGE.
NO
3A. IS THE SESSION NUMBER GREATER THAN THE STORED SESSION NUMBER ?
YES
3B. CPU 46 STORES INFORMATION FROM NEIGHBOR STATUS MESSAGE IN NEW DATA STRUCTURE IN MEMORY AWAITING VALIDATION. IT DISCARDS ANY MESSAGES CURRENTLY STORED. IT SENDS OUT A NEIGHBOR STATUS MESSAGE VIA RING INTERFACE CARDS 30 AND 32 WITH THE NEW SESSION NUMBER.
NO
4A. IS THE SESSION NUMBER EQUAL TO THE STORED SESSION NUMBER ?
YES
4B. IF TOPOLOGY DISCOVERY IS IN PROGRESS, CPU 46 STORES INFORMATION FROM THE NEIGHBOR STATUS MESSAGE IN THE CURRENT DATA STRUCTURE IN MEMORY. OTHERWISE, CPU 46 DISCARDS THE MESSAGE.

FIG. 10

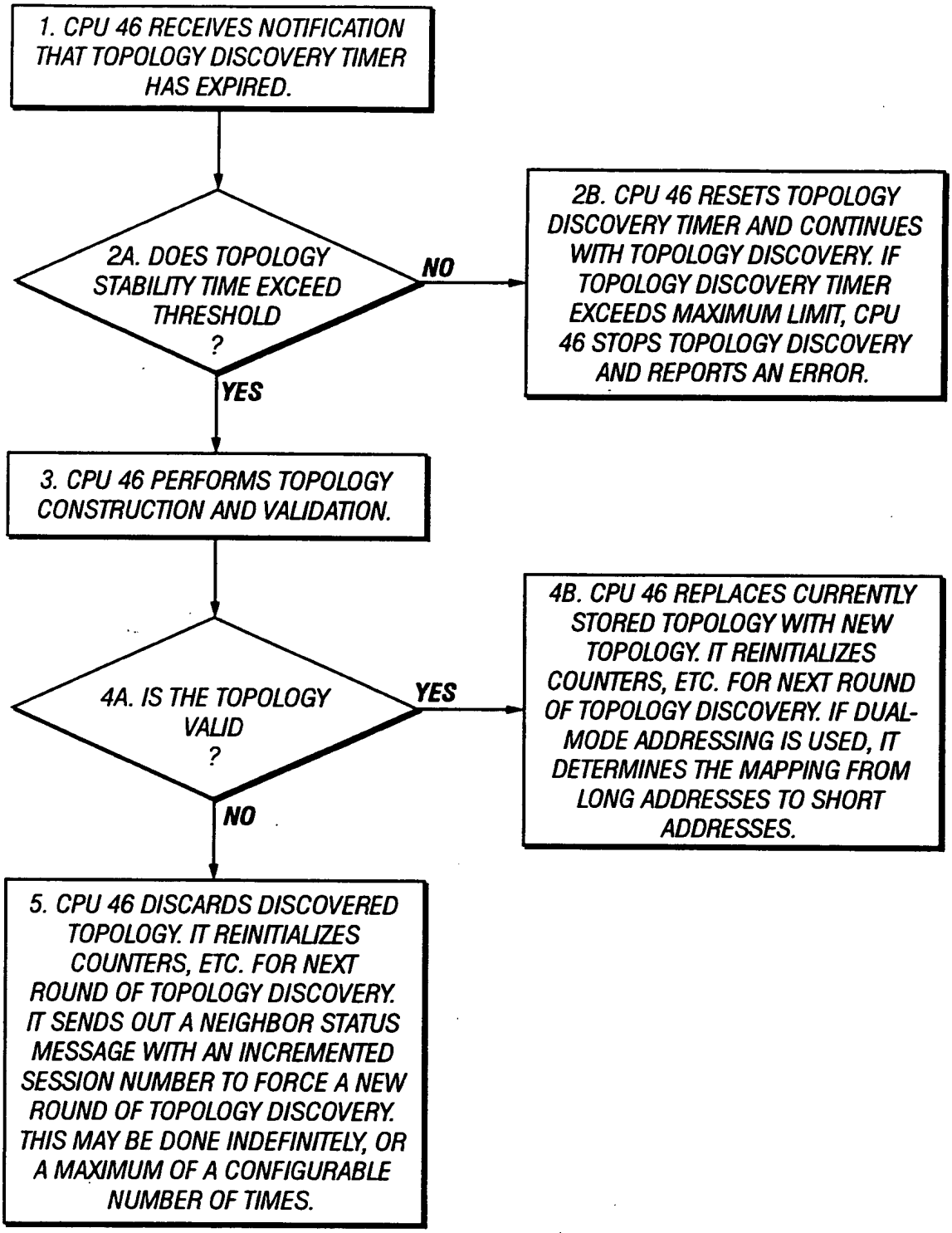
1. CPU 46 RECEIVES NOTIFICATION THAT TOPOLOGY DISCOVERY TIMER HAS EXPIRED.
2A. DOES TOPOLOGY STABILITY TIME EXCEED THRESHOLD ?
NO
2B. CPU 46 RESETS TOPOLOGY DISCOVERY TIMER AND CONTINUES WITH TOPOLOGY DISCOVERY. IF TOPOLOGY DISCOVERY TIMER EXCEEDS MAXIMUM LIMIT, CPU 46 STOPS TOPOLOGY DISCOVERY AND REPORTS AN ERROR.
YES
3. CPU 46 PERFORMS TOPOLOGY CONSTRUCTION AND VALIDATION.
4A. IS THE TOPOLOGY VALID ?
YES
4B. CPU 46 REPLACES CURRENTLY STORED TOPOLOGY WITH NEW TOPOLOGY. IT REINITIALIZES COUNTERS, ETC. FOR NEXT ROUND OF TOPOLOGY DISCOVERY. IF DUAL-MODE ADDRESSING IS USED, IT DETERMINES THE MAPPING FROM LONG ADDRESSES TO SHORT ADDRESSES.
NO
5. CPU 46 DISCARDS DISCOVERED TOPOLOGY. IT REINITIALIZES COUNTERS, ETC. FOR NEXT ROUND OF TOPOLOGY DISCOVERY. IT SENDS OUT A NEIGHBOR STATUS MESSAGE WITH AN INCREMENTED SESSION NUMBER TO FORCE A NEW ROUND OF TOPOLOGY DISCOVERY. THIS MAY BE DONE INDEFINITELY, OR A MAXIMUM OF A CONFIGURABLE NUMBER OF TIMES.

FIG. 11

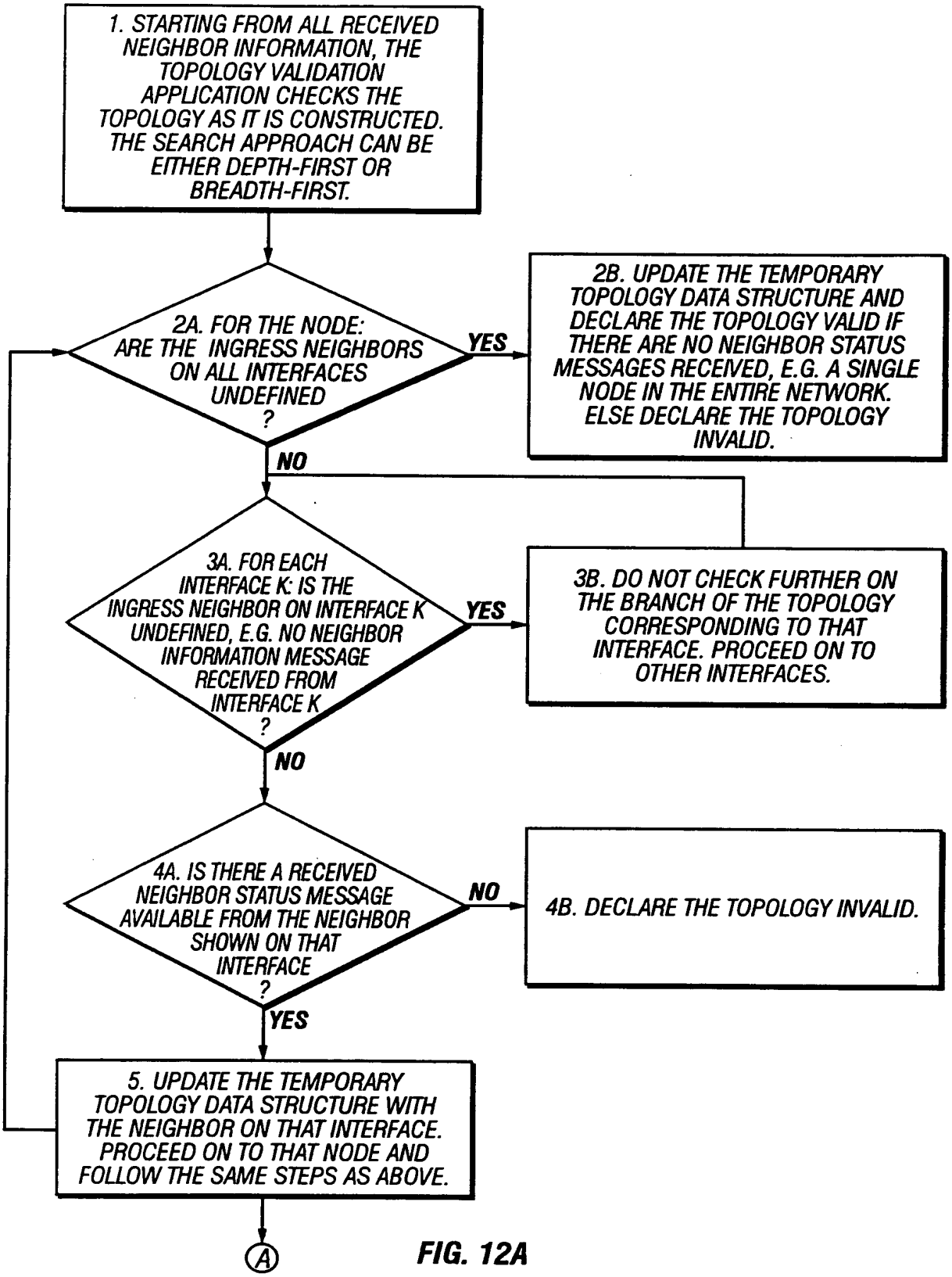
1. STARTING FROM ALL RECEIVED NEIGHBOR INFORMATION, THE TOPOLOGY VALIDATION APPLICATION CHECKS THE TOPOLOGY AS IT IS CONSTRUCTED. THE SEARCH APPROACH CAN BE EITHER DEPTH-FIRST OR BREADTH-FIRST.
2A. FOR THE NODE: ARE THE INGRESS NEIGHBORS ON ALL INTERFACES UNDEFINED ?
YES
2B. UPDATE THE TEMPORARY TOPOLOGY DATA STRUCTURE AND DECLARE THE TOPOLOGY VALID IF THERE ARE NO NEIGHBOR STATUS MESSAGES RECEIVED, E.G. A SINGLE NODE IN THE ENTIRE NETWORK. ELSE DECLARE THE TOPOLOGY INVALID.
NO
3A. FOR EACH INTERFACE K: IS THE INGRESS NEIGHBOR ON INTERFACE K UNDEFINED, E.G. NO NEIGHBOR INFORMATION MESSAGE RECEIVED FROM INTERFACE K ?
YES
3B. DO NOT CHECK FURTHER ON THE BRANCH OF THE TOPOLOGY CORRESPONDING TO THAT INTERFACE. PROCEED ON TO OTHER INTERFACES.
NO
4A. IS THERE A RECEIVED NEIGHBOR STATUS MESSAGE AVAILABLE FROM THE NEIGHBOR SHOWN ON THAT INTERFACE ?
NO
4B. DECLARE THE TOPOLOGY INVALID.
YES
5. UPDATE THE TEMPORARY TOPOLOGY DATA STRUCTURE WITH THE NEIGHBOR ON THAT INTERFACE. PROCEED ON TO THAT NODE AND FOLLOW THE SAME STEPS AS ABOVE.
A

FIG. 12A

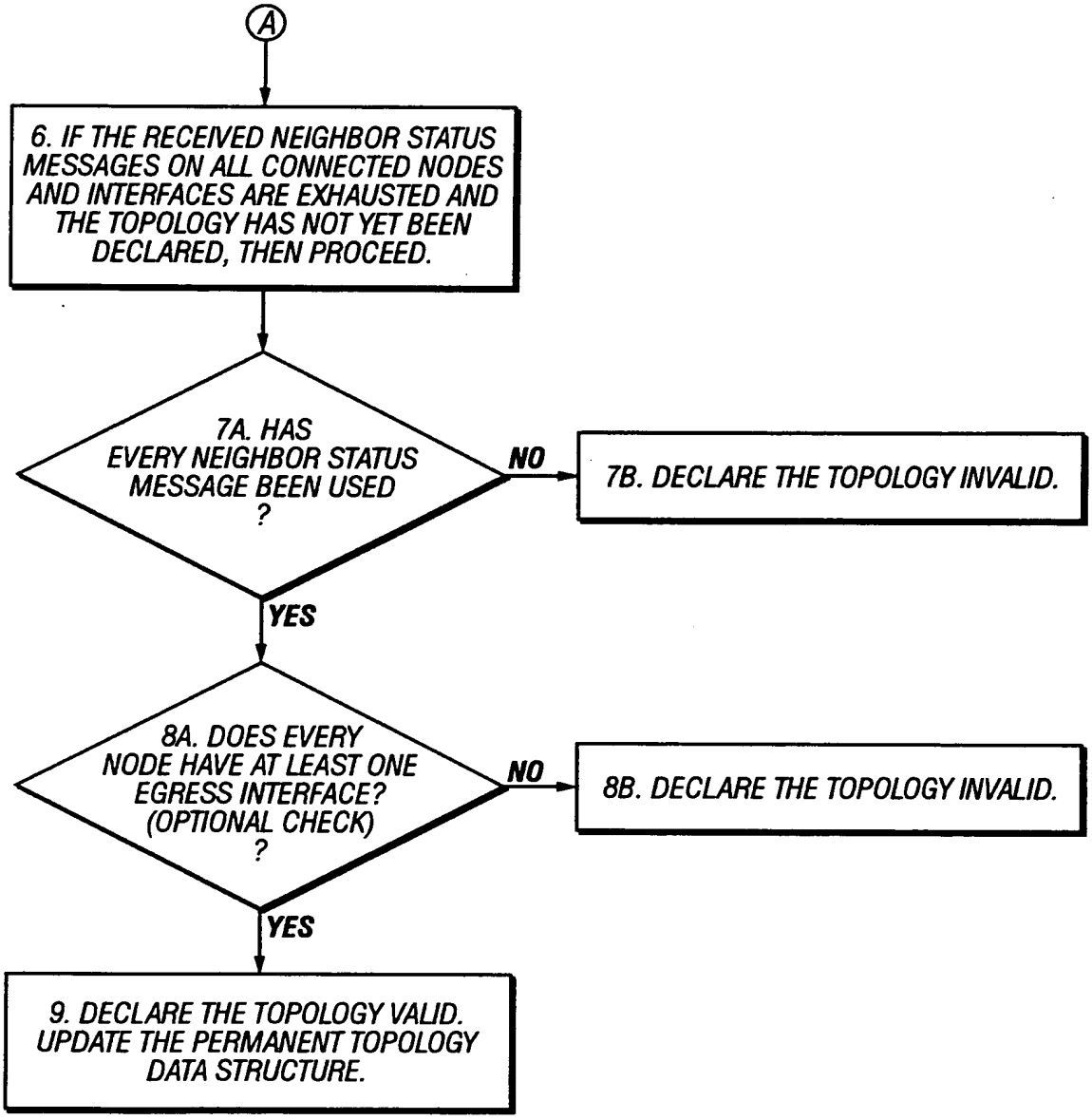
A
6. IF THE RECEIVED NEIGHBOR STATUS MESSAGES ON ALL CONNECTED NODES AND INTERFACES ARE EXHAUSTED AND THE TOPOLOGY HAS NOT YET BEEN DECLARED, THEN PROCEED.
7A. HAS EVERY NEIGHBOR STATUS MESSAGE BEEN USED ?
NO
7B. DECLARE THE TOPOLOGY INVALID.
YES
8A. DOES EVERY NODE HAVE AT LEAST ONE EGRESS INTERFACE? (OPTIONAL CHECK) ?
NO
8B. DECLARE THE TOPOLOGY INVALID.
YES
9. DECLARE THE TOPOLOGY VALID. UPDATE THE PERMANENT TOPOLOGY DATA STRUCTURE.

FIG. 12B